Long Erasure Codes over Burst Erasure Channels: an Optimization Algorithm

M. Chiani, G. Liva and E. Paolini, University of Bologna
mchiani@deis.unibo.it, gliva@deis.unibo.it, epaolini@deis.unibo.it
Outline

I. Introduction and Channel Model

II. Optimization Algorithm Description

III. Numerical Results and Discussion

IV. Conclusions
Introduction

- In wireless communications systems, packet losses at the upper layers can be caused, depending on the scenario and on the transmission band, by weather, shadowing, loss of frame synchronization, or by detection and consequent discarding of erroneously received packets.

- Packet erasures are typically correlated, and they often occur in bursts.

- The memory-less assumption can be correct in some contexts, but it appears quite rough if a packet interleaving procedure (if available) only involves packets belonging to the same codeword, and the erasure burst length is not small respect to the codeword length $n$.

- Long Erasure Codes should be considered over channels where packet erasures occur in bursts.
Objective

• Develop an optimization algorithm able to improve the performance of erasure correcting LDPC codes over channel introducing bursts of erasures

• Desirable features:
  - Efficiency
  - Flexibility (i.e. free from code rate, codeword length, ...)
  - Considerable performance improvement
  - Possibility to use within pure FEC schemes, Type I / Type II ARQ protocols, parity on demand schemes, ...
  - Simplicity
Channel Model

- Constant Length Burst Erasure Channel (CLBuEC)

- A Good state and L Bad states
- \( p_G = 0, p_B = 1 \)
- With probability \( b \) the channel moves from the Good to the Bad states generating an erasure burst of length \( L \)
Optimization Algorithm: the key observation

• Consider an LDPC code with $L'_\text{max}$: at least one erasure burst of length $L'_\text{max}+1$ exists which is non-resolvable.

• Suppose this burst begins on the $j$-th variable node.

- Variable nodes $V_j$ and $V_{j+L'_\text{max}}$ must belong to the maximal stopping set comprised in the burst.

- Non-resolvable burst

- Resolvable burst

- Resolvable burst
Optimization Algorithm: description [1]

• **RULE**: If burst length $L_1$ is non-resolvable due to a burst beginning in position $j$, look for a variable node $V_i$ in the set $\{V_0, \ldots, V_{j-1}, V_{j+L_1}, V_{n-1}\}$ such that permuting $V_j$ (or $V_{j+L_1-1}$) and $V_i$ makes the erasure burst length $L_1$ resolvable.

• Given an input LDPC code with $L_{\text{max}}'$, the algorithm returns a new LDPC code with permuted variable nodes, the same distribution and $L_{\text{max}}'' > L_{\text{max}}'$. 
  
  - Very efficient algorithm (just permutations of variable nodes are performed), applicable for any code rate, very good results.
  
  - Easily combinable with hybrid ARQ schemes, parity on demand, ...

  - If variable nodes $V_i$'s are chosen according to a deterministic rule, the overall algorithm is deterministic.

Optimization Algorithm: random & IRA codes

- **Random Codes (heavy encoding - low practical interest):** The algorithm can be applied on all the encoded symbols.

- **IRA Codes (efficient encoding - high practical interest):** The algorithm can be applied on all the encoded symbols or just on the systematic symbols.
  - All encoded symbols: more powerful (better performance), but a physical interleaver is needed in order to preserve efficient encoding (see next slide).
  - Just systematic symbols: less powerful (worse performance), but IRA structure is preserved (inner LDPC interleaving is optimized and exploited).
Optimization Algorithm: interpretation

- **Algorithm applied on all the IRA encoded symbols:**

  ![Diagram of Optimization Algorithm](image)

  - Powerful but complex solution (interleaver is needed)
  - \( L'''_{\text{max}} \)

- **Algorithm applied on the systematic IRA symbols:**

  ![Diagram of Optimization Algorithm](image)

  - Less powerful but simpler solution (interleaver is not needed)
  - \( L''_{\text{max}} < L'''_{\text{max}} \)
(2000,1000) irregular random LDPC code

- $L_{\text{max}}' = 836$
- $L_{\text{max}}'' = 904$
- $L = 890$

note: $L_{\text{max}}$ asymptotic value is 931
(2000,1000) irregular IRA code

- $L'_\text{max} = 272$
- $L''\text{max} = 621$
- $L'''\text{max} = 900$
- $L = 630, 500$
(1111, 1000) irregular IRA code

- $L'_\text{max} = 46$
- $L''_\text{max} = 79$
- $L'''_\text{max} = 80$
- $L = 70$
Discussion

- R=1/2 random code: improvement in the decoding failure rate of on order of magnitude in terms of b at error rate $10^{-4}$, and even better for lower target performance

- R=1/2 IRA: for $L=630$ ($L_{\text{max}} < L < L_{\text{max}}''$) best performance by applying the algorithm to all the encoded symbols (but the improved IRA is a good tradeoff performance / complexity). For $L=500$ ($L < L_{\text{max}}' < L_{\text{max}}''$) the improved IRA has very good performance as well

- R=9/10 IRA: for high rate codes the influence of the IRA structure is much less evident. The improved IRA has similar performance as original IRA + adapted interleaver

- In general, quite good performance improvement is achieved by the algorithm
Conclusions

• An efficient and simple algorithm has been proposed for improving the performance of LDPC codes when the erasures occur in bursts

• Very high $L_{\text{max}}$'s, near to the asymptotic values, are achieved by the algorithm and consequently substantial improvement in performance is achieved

• Applicable to any LDPC code (example shown for IRA and random codes)

• Since it just improves the original code without any impact on the protocol, the algorithm can be use within pure FEC schemes, Type I / Type II ARQ protocols, parity on demand protocols, etc.